



TUGAS AKHIR -SM141501

# **IMPLEMENTASI *LOSSLESS VISIBLE WATERMARKING* MENGGUNAKAN METODE *COMPOUND MAPPING***

GINANJAR DWI ERDITYAWAN  
NRP 1210 100 054

Dosen Pembimbing  
Dr. Budi Setiyono, S.Si, MT

DEPARTEMEN MATEMATIKA  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**TUGAS AKHIR -SM141501**

**IMPLEMENTASI *LOSSLESS VISIBLE*  
*WATERMARKING* MENGGUNAKAN  
METODE *COMPOUND MAPPING***

**GINANJAR DWI ERDITYAWAN  
NRP 1210 100 054**

**Dosen Pembimbing  
Dr. Budi Setiyono, S.Si, MT**

**DEPARTEMEN MATEMATIKA  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**





FINAL PROJECT -SM141501

***IMPLEMENTATION OF LOSSLESS VISIBLE  
WATERMARKING USING COMPOUND MAPPING  
METHOD***

GINANJAR DWI ERDITYAWAN  
NRP 1210 100 054

Supervisor  
Dr. Budi Setiyono, S.Si, MT

MATHEMATICS DEPARTMENT  
Faculty of Mathematics and Natural Science  
Sepuluh Nopember Institute of Technology  
Surabaya 2017



# LEMBAR PENGESAHAN

## IMPLEMENTASI *LOSSLESS VISIBLE* WATERMARKING MENGGUNAKAN METODE *COMPOUND MAPPING*

### *IMPLEMENTATION OF LOSSLESS VISIBLE WATERMARKING USING COMPOUND MAPPING METHOD*

#### TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat  
Untuk memperoleh gelar Sarjana Sains  
Pada bidang studi Ilmu Komputer  
Program Studi S-1 Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

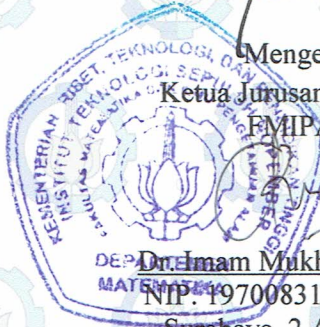
GINANJAR DWI ERDITYAWAN  
NRP. 1210 100 054

Menyetujui,  
Dosen Pembimbing,

Dr. Budi Setiyono, S.Si, MT  
NIP. 19720207 199702 1 001

Mengetahui,  
Ketua Jurusan Matematika  
FMIPA ITS

Dr. Imam Mukhlash, S.Si, MT  
NIP. 19700831 199403 1 003  
Surabaya, 2 Agustus 2017







# ***IMPLEMENTATION OF LOSSLESS VISIBLE WATERMARKING USING COMPOUND MAPPING METHOD***

***Name*** : ***Ginanjari Dwi Erdityawan***  
***NRP*** : ***1210 100 054***  
***Department*** : ***Mathematics***  
***Supervisor*** : ***Dr. Budi Setiyono, S.Si, MT***

## ***Abstract***

*Distribution of digital information is made easier by the development of computer technology and internet. That development raised a new problem called violation of right of digital creation, especially image. To tackle the problem, a technique called digital watermarking is used. In this final project, a method called Compound Mapping is used as a one of reversible watermarking technique. At first, an image, as a watermark, will be put to another image. Then the watermarked image will be cleaned (erasing the watermark) to get recovered image which have same pixel data as the original image. Based on the test, the system is able to handle insertion of watermark well. System is able to handle the recovery of watermarked image, PNG and BMP format, without noise well, which is exactly the same as the original image, proved by zero MSE test. However, the system is not able to recover a watermarked JPG image well, which is the recovered image is not exactly the same as the original image. Also the system is not be able to recover a noised watermarked image to get the exact same image as the original image.*

***Keywords:*** *Watermarking, Reversible Watermarking, Lossless Visible Watermarking, Compound Mapping Method, One to One Compound Mapping.*



# **IMPLEMENTASI *LOSSLESS VISIBLE WATERMARKING* MENGGUNAKAN METODE *COMPOUND MAPPING***

**Nama Mahasiswa : Ginanjar Dwi Erdityawan**  
**NRP : 1210 100 054**  
**Departemen : Matematika**  
**Dosen Pembimbing : Dr. Budi Setiyono, S.Si, MT**

## **Abstrak**

Berkembangnya teknologi komputer dan internet, menyebabkan pendistribusian informasi digital semakin mudah. Hal tersebut menimbulkan masalah baru diantaranya pelanggaran hak cipta dan penyalahgunaan karya digital. Untuk mengatasi permasalahan tersebut digunakan teknik *digital watermarking*. Pada tugas akhir ini diimplementasikan metode *Compound Mapping* untuk digunakan dalam teknik *reversible watermarking*. Prosesnya adalah citra objek disisipi dengan citra lain sebagai *watermark* untuk menghasilkan citra yang ter-*watermark*. Selanjutnya citra yang telah ter-*watermark* tersebut dipulihkan (menghilangkan *watermark*) sehingga menghasilkan citra yang memiliki kesamaan data piksel dengan citra objek awal. Berdasarkan hasil pengujian, sistem yang dibangun mampu menangani proses penyisipan *watermark* dengan baik. Untuk pemulihan citra ter-*watermark* tanpa *noise* dengan format PNG dan BMP, sistem mampu menghasilkan citra yang sama persis dengan citra awal, terbukti dengan hasil uji MSE yang bernilai nol. Sedangkan untuk citra ter-*watermark* berformat JPG hasil pemulihannya tidak sama persis dengan citra awal. Untuk citra ter-*watermark* yang diberi *noise*, pemulihannya tidak menghasilkan citra yang sama persis dengan citra awal.

***Kata Kunci***—*Watermarking, Reversible Watermarking, Lossless Visible Watermarking, Metode Compound Mapping, One to One Compound Mapping.*



## KATA PENGANTAR

Segala Puji bagi Allah SWT yang telah memberikan karunia, rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul: **“Implementasi Lossless Visible Watermarking Menggunakan Metode Compound Mapping”** yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Studi S1 pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan berkat kerjasama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal itu, penulis mengucapkan terima kasih kepada:

1. Ketua Departemen Matematika ITS.
2. Ketua Program Studi S1 Jurusan Matematika ITS
3. Sekertaris Program Studi S1 Jurusan Matematika ITS
4. Dra. Titik Mudjiati, M. Si selaku Dosen Wali.
5. Dr. Budi Setiyono, S.Si, MT selaku dosen pembimbing yang senantiasa membimbing dengan sabar dan memberikan kritik dan saran dalam penyusunan Tugas Akhir ini.
6. Dr. Mahmud Yunus, M.Si, Drs. Nurul Hidayat, M.Kom dan Dr. Dwi Ratna Sulistyaningrum, S.Si, MT selaku dosen penguji Tugas Akhir ini.
7. Seluruh jajaran dosen dan staf jurusan Matematika ITS.
8. Kedua orang tua dan kakak yang selalu memberikan doa dan dukungan kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan saran dan kritik dari pembaca. Akhir kata, semoga Tugas Akhir ini bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2017

**Penulis**



## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	v
<b>ABSTRAK</b> .....	vii
<b>ABSTRACT</b> .....	ix
<b>KATA PENGANTAR</b> .....	xi
<b>DAFTAR ISI</b> .....	xiii
<b>DAFTAR GAMBAR</b> .....	xv
<b>DAFTAR TABEL</b> .....	xvii
 <b>BAB I. PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Manfaat .....	3
1.6 Sistematika Penulisan Tugas Akhir .....	4
 <b>BAB II. TINJAUAN PUSTAKA</b>	
2.1 Pengertian Citra .....	7
2.2 Citra Digital .....	8
2.2.1 Representasi Citra Digital .....	8
2.3 <i>Digital Watermarking</i> .....	9
2.3.1 <i>Reversible Watermarking</i> .....	9
2.4 <i>Compound Mapping</i> .....	10
2.4.1 <i>Reversible One to One Compound Mapping</i> .....	10
2.4.2 <i>Lossless Visible Watermarking</i> .....	11
2.5 <i>Salt and Pepper Noise</i> .....	12
2.6 <i>Mean Square Error</i> .....	12
 <b>BAB III. METODOLOGI PENELITIAN</b>	
3.1 Objek Penelitian.....	13
3.2 Peralatan .....	13
3.3 Tahap Penelitian .....	13

#### **BAB IV. LOSSLESS VISIBLE WATERMARKING DENGAN METODE ONE TO ONE COMPOUND MAPPING**

4.1	Analisis Sistem.....	17
4.1.1	Deskripsi Sistem .....	17
4.2	Perancangan Sistem .....	19
4.2.1	Perancangan Penyisipan Citra <i>Watermark</i> ..	19
4.2.2	Perancangan Pemulihan Citra <i>Watermark</i> .	26
4.2.3	Kasus Khusus.....	33
4.2.4	Perancangan Antar Muka Sistem.....	36
4.3	Implementasi Sistem.....	37
4.3.1	Implementasi <i>Input</i> Citra .....	37
4.3.2	Implementasi Pemilihan Area <i>Watermark</i> .	38
4.3.3	Implementasi Proses <i>Resizing</i> Citra <i>Watermark</i> .....	39
4.3.4	Implementasi Proses Penyisipan Citra <i>Watermark</i> .....	39
4.3.5	Implementasi Pemilihan Area Hasil.....	40
4.3.6	Implementasi Proses Pemulihan Citra ter- <i>Watermark</i> .....	41

#### **BAB V. UJI COBA DAN ANALISIS HASIL**

5.1	Uji Coba Terhadap Citra Tanpa <i>Noise</i> .....	43
5.2	Uji Coba Terhadap Citra dengan- <i>Noise</i> .....	49

#### **BAB VI. KESIMPULAN DAN SARAN**

6.1	Kesimpulan .....	55
6.2	Saran .....	56

<b>DAFTAR PUSTAKA</b> .....	57
-----------------------------	----

<b>BIODATA PENULIS</b> .....	59
------------------------------	----



## DAFTAR GAMBAR

	Halaman
Gambar 1.1	Properti Interior yang Diberi <i>Watermark</i> ..... 2
Gambar 2.1	Matriks Citra <i>Digital</i> ..... 8
Gambar 2.2	<i>Visible Watermarking</i> pada Citra Lena..... 9
Gambar 2.3	Diagram <i>Reversible Watermark</i> ..... 10
Gambar 3.1	Blok Diagram <i>Watermarking</i> Citra dengan Metode <i>One to one Compound Mapping</i> ..... 16
Gambar 4.1	Diagram Alir Sistem ..... 18
Gambar 4.2	Diagram Alir Penyisipan Citra <i>Watermark</i> ..... 19
Gambar 4.3	Contoh Citra <i>Host</i> ..... 20
Gambar 4.4	Ilustrasi <i>Resizing</i> Citra <i>Watermark</i> ..... 21
Gambar 4.5	Ilustrasi Citra. <i>I</i> adalah citra <i>host</i> . <i>L</i> adalah citra <i>watermark</i> . <i>W</i> adalah citra ter- <i>watermark</i> . ..... 23
Gambar 4.6	Ilustrasi persekitaran piksel. Titik yang dilingkari merah merupakan persekitaran dari piksel p..... 24
Gambar 4.7	Pengambilan segmen citra untuk penghitungan..... 24
Gambar 4.8	Nilai piksel dari masing-masing segmen citra yang diambil pada Gambar 4.7. <i>I</i> adalah citra <i>host</i> . <i>L</i> adalah citra <i>watermark</i> . <i>W</i> adalah citra ter- <i>watermark</i> . ..... 25
Gambar 4.9	Ilustrasi Perhitungan Nilai Piksel ..... 26
Gambar 4.10	Citra ter- <i>Watermark</i> Hasil Proses Penyisipan..... 26
Gambar 4.11	Diagram Alir Pemulihan Citra ter- <i>Watermark</i> ..... 27
Gambar 4.12	Ilustrasi Citra. <i>W</i> adalah citra ter- <i>watermark</i> . <i>L</i> adalah citra <i>watermark</i> . <i>R</i> adalah citra output (hasil) ..... 29
Gambar 4.13	Hasil Pemetaan Nilai Piksel di Luar Area <i>Watermark</i> ..... 29
Gambar 4.14	Pengambilan segmen citra untuk penghitungan ..... 30
Gambar 4.15	Nilai piksel dari masing-masing segmen citra yang diambil pada Gambar 4.14. <i>W</i> adalah citra ter- <i>watermark</i> . <i>L</i> adalah citra <i>watermark</i> . <i>R</i> adalah citra output (hasil pemulihan) ..... 31
Gambar 4.16	Citra <i>R</i> setelah ditentukan nilai piksel pertama pada area hasil P ..... 32
Gambar 4.17	Ilustrasi perhitungan nilai piksel ..... 32
Gambar 4.18	Contoh Citra Hasil Pemulihan ..... 32
Gambar 4.19	Contoh Nilai Piksel..... 33

Gambar 4.20	Nilai Piksel Setelah Dilakukan Pemetaan dari $P$ ke $Q$ .....	34
Gambar 4.21	Nilai Piksel Setelah Dilakukan Pemetaan di Luar Area Hasil .....	34
Gambar 4.22	Nilai Piksel Setelah Dilakukan Pemetaan dari $Q$ ke $P$ .....	35
Gambar 4.23	Nilai Piksel Citra <i>Host</i> ( $I$ ) dan Citra Hasil Pemulihan ( $R$ ) .....	35
Gambar 4.24	Antarmuka Halaman Program <i>Watermarking</i> .....	37
Gambar 4.25	Tampilan saat pemilihan citra <i>input</i> .....	38
Gambar 4.26	Tampilan saat pemilihan area <i>watermark</i> .....	38
Gambar 4.27	Implementasi proses penyisipan citra <i>watermark</i> ....	40
Gambar 4.28	Tampilan saat pemilihan area hasil .....	40
Gambar 4.29	Antar muka proses pemulihan citra ter- <i>watermark</i> .	41
Gambar 5.1	Citra <i>buildings.png</i> ter- <i>Watermark</i> .....	50
Gambar 5.2	Grafik Nilai MSE Pada Uji Citra ber- <i>Noise</i> .....	54

## DAFTAR TABEL

	Halaman
Tabel 5.1 Data Citra <i>Host</i> yang Digunakan .....	44
Tabel 5.2 Data Citra <i>Watermark</i> yang Digunakan.....	45
Tabel 5.3 Data Citra ter- <i>Watermark</i> .....	45
Tabel 5.4 Data Citra Hasil Pemulihan dan Nilai MSE .....	47
Tabel 5.5 Data Citra ter- <i>Watermark</i> ber- <i>Noise</i> .....	50
Tabel 5.6 Data Citra Hasil Pemulihan dari Citra ber- <i>Noise</i> dan Nilai MSE .....	52



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Semakin berkembangnya teknologi komputer dan internet, menyebabkan pendistribusian informasi digital menjadi semakin mudah. Namun hal tersebut juga menimbulkan suatu masalah baru yaitu sulitnya mendapatkan hak cipta untuk suatu karya digital karena data yang ada di internet bisa didapatkan dengan begitu mudah. Banyak hasil karya yang dengan mudah digunakan oleh pengguna internet tanpa mendapatkan ijin dari pemilik karya yang asli, bahkan juga banyak karya yang disalahgunakan atau diakui menjadi milik orang lain.

Karena permasalahan diatas, dibutuhkan suatu teknik tertentu untuk menandai hasil karya digital agar tidak dapat digunakan secara sembarangan oleh orang lain yang bukan pemilik karya tersebut. Teknik yang bisa digunakan untuk mengatasi permasalahan tersebut adalah *digital watermarking*. *Digital Watermarking* adalah proses penyisipan informasi ke dalam data digital sehingga informasi tambahan (yang disebut sebagai *watermark*) tersebut dapat diekstraksi dan dideteksi untuk berbagai tujuan termasuk pencegahan dan pengendalian dalam masalah penggandaan data [2]. Teknik ini bekerja dengan cara menambahkan informasi pada gambar atau video yang dimaksud dengan tujuan untuk memberikan tanda tertentu sehingga gambar atau video yang diunggah ke internet dapat dipertanggungjawabkan penggunaannya.

Pada beberapa data gambar di internet, contohnya pada *web* perusahaan yang bergerak di bidang desain, perlu untuk menampilkan contoh-contoh desain buatan perusahaan tersebut. Namun desain yang ditampilkan juga perlu diamankan agar tidak

disalahgunakan oleh pihak lain, sehingga perlu diberi *watermark* seperti pada Gambar 1.1 [7]. Di sisi lain, gambar tersebut juga dibutuhkan oleh pihak-pihak yang memang membutuhkan dan berhak untuk mendapatkan gambar tersebut, seperti para agen yang tugasnya adalah menawarkan desain tersebut kepada para pelanggan. Atas alasan tersebut, maka dibutuhkan suatu teknik pemulihan untuk gambar yang telah diberi *watermark* agar gambar hasil pemulihannya sama dengan gambar aslinya.



**Gambar 1.1** Properti Interior yang diberi Watermark

Oleh karena itu pada tugas akhir ini diimplementasikan metode *Compound Mapping* untuk digunakan dalam teknik *reversible watermarking*. Citra objek akan disisipi dengan citra lain sebagai *watermark* untuk menghasilkan citra yang ter-*watermark*. Proses penyisipan tersebut dilakukan dengan metode *compound mapping*. Selanjutnya citra yang telah ter-*watermark* tersebut dipulihkan (menghilangkan *watermark*) dengan metode yang sama sehingga menghasilkan citra yang memiliki kesamaan data piksel dengan citra objek awal.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka permasalahan yang akan dibahas pada Tugas Akhir ini adalah :

1. Bagaimana mengimplementasikan *reversible watermarking* pada citra *digital* menggunakan metode komposisi pemetaan satu-satu.
2. Bagaimana menguji *watermarking* dengan *noise*.

### 1.3 Batasan Masalah

Batasan Masalah yang akan dibahas dalam penelitian tugas akhir ini adalah sebagai berikut :

1. Citra yang menjadi *watermark* hanya satu citra.
2. Citra *watermark* disisipkan seluruhnya ke dalam citra *host*.
3. Citra *watermark* dan citra asli yang digunakan untuk uji coba adalah citra bertipe BMP, PNG, dan JPG.
4. Model *noise* yang diberikan adalah *noise salt and pepper*.
5. Pengujian kesamaan citra dilakukan dengan uji MSE.
6. Pengerjaan program *watermarking* pada penelitian ini menggunakan *software* matlab.

### 1.4 Tujuan

Tujuan dari tugas akhir ini adalah :

1. Mengimplementasikan *reversible watermarking* pada citra *digital* menggunakan metode komposisi pemetaan satu-satu.
2. Menguji *watermarking* dengan *noise*.

### 1.5 Manfaat

Manfaat dari tugas akhir ini adalah sebagai berikut :

1. Mampu mengamankan citra digital dari penggunaan maupun penyalahgunaan oleh pihak lain.
2. Dapat digunakan agen marketing untuk memasarkan produknya dalam bentuk citra *digital*.
3. Sebagai tambahan kepustakaan untuk penelitian selanjutnya.

## **1.6 Sistematika Penulisan Tugas Akhir**

Sistematika penulisan didalam Tugas Akhir ini adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Pada bab ini akan dibahas mengenai latar belakang dari tugas akhir ini, permasalahan yang akan dibahas dan diselesaikan, batasan masalah yang digunakan dalam menyelesaikan permasalahan yang telah dirumuskan, serta tujuan dan manfaat yang ingin dicapai dalam Tugas Akhir ini.

### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi penjelasan mengenai kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam tugas akhir ini.

### **BAB III METODOLOGI PENELITIAN**

Pada bab ini akan dijelaskan metodologi atau urutan pengerjaan yang dilakukan dalam menyelesaikan tugas akhir dimana langkah yang dilakukan meliputi studi literatur, perancangan dan implementasi perangkat lunak, uji coba dan pembahasan, hingga penarikan kesimpulan.

### **BAB IV PERANCANGAN DAN IMPLEMENTASI**

Bab ini menjelaskan perancangan gambaran umum sistem, perancangan proses algoritma, dan perancangan antar muka sistem. Sistem ini membutuhkan dua buah inputan masing-masing sebagai citra *host* dan citra *watermark*. Selanjutnya dilakukan proses penyisipan citra *watermark* ke dalam citra *host* sehingga menghasilkan citra ter-*watermark*. Kemudian citra ter-*watermark* akan diberi tambahan *noise* untuk menguji kehandalan *watermark*-nya. Selanjutnya dilakukan proses ekstraksi sehingga menghasilkan citra yang sama



dengan citra asli. Hasil tersebutlah yang akan digunakan dalam proses pengujian.

## BAB V PENGUJIAN DAN PEMBAHASAN HASIL

Bab ini akan menampilkan hasil dari proses ekstraksi serta nilai *mean square error* (MSE) sebagai hasil pengujian kesamaan citra hasil ekstraksi dengan citra objek awal. Hasil pengujian inilah yang akan digunakan dalam perumusan kesimpulan dan saran.

## BAB VI PENUTUP

Bab ini merupakan penutup, berisi tentang kesimpulan yang dapat diambil berdasarkan uji coba yang telah dilakukan dan saran yang selayaknya dilakukan bila tugas akhir ini dilanjutkan.



## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini dijelaskan tentang kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam tugas akhir ini, meliputi Pengertian Citra *Digital*, *Digital Watermarking*, Metode *One to one Compound Mapping* untuk proses penyisipan serta pemulihan citra ter-watermark, MSE (*Mean Square Error*) untuk menguji kualitas citra awal dan citra akhir.

#### **2.1 Pengertian Citra**

Citra menurut Kamus Besar Bahasa Indonesia memiliki makna rupa, gambar, atau gambaran. Sedangkan menurut kamus *Webster* citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Citra terbagi menjadi dua yaitu citra diam dan citra bergerak. Citra diam adalah citra tunggal yang tidak bergerak. Sedangkan, citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan pada mata kita sebagai gambar yang bergerak.

Dalam beberapa masa, citra yang dikenal manusia berbentuk citra kontinu. Suatu representasi objek yang dihasilkan dari sistem optik yang menerima sinyal analog dan dinyatakan dalam bidang dua dimensi. Nilai cahaya yang ditransmisikan pada citra kontinu memiliki rentang nilai yang tak terbatas. Contoh dari citra kontinu adalah mata manusia dan kamera analog.

Sebuah citra analog tidak dapat direpresentasikan secara langsung oleh komputer. Oleh sebab itu dilakukan sebuah proses untuk merubah nilai-nilai yang ada pada citra analog agar komputer dapat membaca dan menerjemahkan informasi yang terdapat pada citra analog. Hasil dari pemrosesan tersebut dinamakan sebagai citra *digital*.

## 2.2 Citra Digital

Citra *digital* merupakan fungsi dua dimensi yang dapat dinyatakan dengan fungsi  $f(x,y)$ , dimana  $x$  dan  $y$  merupakan titik koordinat spasial. Dan amplitudo dari fungsi  $f$  pada sembarang koordinat  $(x,y)$  merupakan nilai intensitas cahaya, yang merupakan representasi dari warna cahaya yang ada pada citra analog. Citra *digital* adalah suatu citra dimana  $(x,y)$  dan nilai intensitas dari  $f$  terbatas (*discrete quantities*), dan telah dilakukan proses digitalisasi spasial dan digitalisasi kuantitas [3].

### 2.2.1 Representasi Citra Digital

Citra *digital* diperoleh dari proses digitalisasi citra, yaitu proses pengubahan citra dari fungsi kontinu menjadi fungsi diskrit. Secara umum, citra *digital* berbentuk segi empat dengan dimensi ukurannya dinyatakan oleh panjang dan lebar. Misal suatu citra dengan panjang  $M$  dan lebar  $N$ , dinyatakan dengan bentuk matriks yang berukuran  $M \times N$  sebagai berikut [3]:

$$\begin{Bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{Bmatrix}$$

**Gambar 2.1** Matriks Citra Digital

Pada representasi *digital* dari citra *grayscale* (abu-abu), citra disajikan dalam bilangan *array* dua dimensi, tiap nilai mewakili intensitas atau level *gray* dari citra tersebut. Jika tiap level direpresentasikan oleh 8 bits (1 *byte*), maka level *gray* dapat dituliskan  $2^8$  atau 256 kemungkinan nilai. Level ini biasanya ditandai dengan nilai *integer* dari 0 sampai 255, dengan nilai 0 mewakili level paling gelap dan nilai 255 mewakili level paling terang. Tiap elemen dalam matriks disebut elemen gambar yang biasa disebut piksel. Dalam citra berwarna, representasinya sama, kecuali tiap nilainya mewakili tiga warna utama yaitu, merah

(*red*), hijau (*green*), dan biru (*blue*). Untuk 24 bit representasi warna per piksel, nilainya dibagi menjadi tiga *channel* 8 bit. Setiap *channel* mewakili intensitas dari salah satu warna utama.

### 2.3 Digital Watermarking

*Digital Watermarking* adalah proses penyisipan informasi kedalam data digital sehingga informasi tambahan (yang disebut sebagai *watermark*) tersebut dapat diekstraksi dan dideteksi untuk berbagai tujuan termasuk pencegahan dan pengendalian dalam masalah penggandaan data [2]. *Digital watermarking* terbagi atas dua tipe yaitu, *invisible watermarking* dan *visible watermarking*. *Invisible watermarking* adalah *watermark* yang hasilnya tidak dapat dilihat dengan mata telanjang. *Invisible watermarking* biasanya digunakan untuk penyisipan informasi hak cipta untuk mencegah pelanggaran hak cipta [4]. Keberadaan *invisible watermarking* hanya dapat dilihat melalui ekstraksi *watermark* atau menggunakan algoritma tertentu. Sedangkan *visible watermarking* adalah sub-citra atau citra semi-transparan (*opaque*) yang diletakkan diatas citra yang lain (citra yang diberi *watermark*), sehingga *watermark*-nya masih dapat dilihat oleh mata telanjang. Contoh hasil *visible watermark* ditunjukkan pada Gambar 2.2 [4].

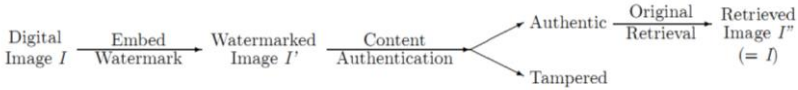


**Gambar 2.2** *Visible Watermarking* pada Citra Lena

#### 2.3.1 Reversible Watermarking

*Reversible Watermarking* adalah *Digital Watermarking* dengan fitur yang menarik yaitu mampu menghilangkan

*watermark* hanya pada konten yang terautentikasi [5]. Misalkan dilakukan penyisipan *reversible watermark* pada citra *digital*  $I$ , sehingga diperoleh citra ter-*watermark*  $I'$ . Sebelum mengirimnya ke konten autentikator, mungkin saja terjadi kerusakan terhadap citra  $I'$ , baik secara sengaja maupun tidak. Jika autentikator menyatakan bahwa tidak terjadi kerusakan pada citra  $I'$ , dengan kata lain citra  $I'$  adalah konten autentik, maka autentikator akan menghapus *reversible watermark* pada  $I'$  dan diperoleh citra seperti yang semula, yang disebut sebagai  $I''$ . Berdasarkan definisi *reversible watermark*, citra yang diperoleh,  $I''$ , akan sama persis dengan citra asli  $I$ , piksel per piksel. Diagram *reversible watermarking* ditunjukkan pada Gambar 2.3 [6].



**Gambar 2.3** Diagram *Reversible Watermark*

## 2.4 Compound Mapping

### 2.4.1 Reversible One to One Compound Mapping

Definisikan sebuah komposisi pemetaan satu-satu  $f$  untuk mengkonversi sekumpulan nilai numerik  $P = \{p_1, p_2, \dots, p_m\}$  ke sekumpulan nilai lain  $Q = \{q_1, q_2, \dots, q_m\}$  sehingga pemetaan dari  $p_i$  ke  $q_i$ , untuk setiap  $i = 1, 2, \dots, m$  adalah *reversible*. Semua nilai  $p_i$  dan  $q_i$  adalah nilai piksel dari sebuah citra (*grayscale* atau berwarna). Komposisi pemetaan  $f$  diatur oleh fungsi satu-satu  $F_x$  dengan satu parameter  $x = a$  atau  $b$  sebagai berikut :

$$q = f(p) = F_b^{-1}(F_a(p)) \quad (2.1)$$

dimana  $F_x^{-1}$  adalah invers dari  $F_x$  dimana dengan menggunakan properti satu-satu, mengarah pada fakta bahwa jika  $F_a(p) = p'$  maka  $F_a^{-1}(p') = p$  untuk setiap  $a$  dan  $p$ . Di sisi lain,  $F_a(p)$  dan  $F_b(p)$  secara umum akan berbeda jika  $a \neq b$ .

Komposisi pemetaan seperti yang telah dideskripsikan pada persamaan (2.1) tentunya *reversible*, yaitu  $p$  dapat ditentukan dari  $q$  menggunakan persamaan 2.2

$$p = f^{-1}(q) = F_a^{-1}(F_b(q)) \quad (2.2)$$

**Lemma 1.**

Jika  $q = F_b^{-1}(F_a(p))$  untuk setiap fungsi satu-satu  $F_x$  dengan parameter  $x$ , maka  $p = F_a^{-1}(F_b(q))$  untuk setiap nilai  $a$ ,  $b$ ,  $p$ , dan  $q$ .

**2.4.2 Lossless Visible Watermarking**

Berdasarkan Lemma 1, akan diturunkan usulan skema *generic lossless visible watermarking* dalam bentuk komposisi pemetaan satu-satu, yang dapat digunakan untuk penyisipan bermacam *visible watermark* ke dalam citra. Penyisipan tersebut bersifat *reversible*, yaitu *watermark* dapat dihilangkan untuk mengembalikan citra asli secara *lossless*. Untuk tujuan ini, didefinisikan lemma 2 di bawah ini:

**Lemma 2.**

Sangat memungkinkan untuk menggunakan komposisi pemetaan  $q = F_b^{-1}(F_a(p))$  untuk mengkonversikan sebuah nilai numerik  $p$  ke dalam nilai lain yang mendekati nilai  $l$  yang diinginkan.

Bukti dari lemma di atas adalah sebagai berikut. Diberikan  $F_x(p) = p - x$ , dimana  $x$  adalah parameter untuk  $F$ . Lalu  $F_x^{-1}(p') = p' + x$ . Lalu diberikan  $a = p - \varepsilon$  dan  $b = l$ , sehingga komposisi pemetaan  $F_b^{-1}(F_a(p))$  menghasilkan

$$\begin{aligned} q &= F_b^{-1}(F_a(p)) \\ &= F_b^{-1}(p - a) \\ &= F_b^{-1}(\varepsilon) \\ &= \varepsilon + b \\ &= \varepsilon + l \end{aligned}$$

yang berarti nilai  $q$  mendekati nilai  $l$ .

***Teorema 1 (Lossless Reversible Visible Watermarking)***

Terdapat komposisi pemetaan satu-satu yang bisa digunakan untuk penyisipan *visible watermark*  $Q$  ke dalam citra  $I$ , dimana nilai pikselnya mendekati *watermark*  $L$ , sehingga citra asli  $I$  dapat dipulihkan dari  $Q$  secara *lossless*.

**2.5 *Salt and Pepper Noise***

*Noise* ini disebut juga dengan *Data Drop Noise* karena secara statistik *noise* ini menurunkan (*drop*) nilai data yang asli [1]. Namun, tidak sepenuhnya citra yang terkena *noise salt and pepper* mengalami penurunan nilai data, hanya beberapa piksel saja yang mengalami penurunan nilai. *Noise* jenis ini terjadi pada saat transmisi data. Nilai piksel citra terganti dengan nilai maksimal atau minimal. Missal untuk citra 8 bit, nilai maksimal adalah 255 dan nilai minimal adalah 0.

**2.6 *Mean Square Error ( MSE )***

MSE adalah metode pengukuran kontrol dan kualitas yang diperoleh dari perhitungan perbandingan contoh obyek dengan obyek asli sehingga diketahui tingkat ketidaksesuaiannya. Persamaan MSE terhadap deviasi target dirumuskan sebagai berikut :

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x, y) - I'(x, y)]^2 \quad (2.3)$$

Dengan  $I(x, y)$  adalah nilai piksel asli, sedangkan  $I'(x, y)$  adalah nilai piksel citra hasil rekonstruksi, dan  $M, N$  adalah dimensi citra.



## **BAB III**

### **METODOLOGI PENELITIAN**

Bab ini membahas mengenai metodologi sistem yang digunakan untuk menyelesaikan tugas akhir. Pembahasan metodologi sistem diawali dengan penjelasan tentang objek penelitian, peralatan yang digunakan, dan tahap penelitian.

#### **3.1 Objek Penelitian**

Objek penelitian yang akan digunakan pada tugas akhir adalah citra berwarna yang berlaku sebagai citra *host* serta citra berwarna yang lebih kecil yang berlaku sebagai *watermark*.

#### **3.2 Peralatan**

Peralatan penelitian yang digunakan untuk menyelesaikan tugas akhir ini adalah MATLAB sebagai perangkat lunak utama untuk membuat antarmuka dan membangun sistem *watermarking*.

#### **3.3 Tahap Penelitian**

Adapun tahap-tahap yang dilakukan dalam penyusunan Tugas Akhir ini adalah sebagai berikut :

1. Studi Literatur

Pada tahap ini dilakukan pengkajian mengenai *digital watermarking* dan metode-metode yang dapat digunakan untuk menghasilkan *visible watermarking* yang *reversible* dan hasil pemulihannya mampu mendekati citra asli. Studi ini dilakukan dengan membaca jurnal *visible watermarking* yang menggunakan *compound mapping*, serta jurnal lain yang berkaitan dengan *reversible watermarking*.

2. Analisis dan Perancangan Sistem

Pada tahapan ini akan dilakukan analisis terhadap sistem yang akan dibuat dan dilakukan pula perancangan proses algoritma sistem meliputi dua proses utama yaitu proses

penyisipan citra *watermark* dan proses pemulihan citra ter-*watermark*. Berikut adalah prosedur dalam sistem *reversible visible watermarking* :

- (a) *Input* citra *host* yang merupakan masukan utama dalam sistem ini karena nantinya akan disisipi dengan *watermark*.
- (b) Lakukan pemilihan area *watermark* pada citra *host* dimana area ini merupakan area yang akan disisipi *watermark*.
- (c) *Input* citra *watermark* yang akan disisipkan ke citra *host*.
- (d) Lakukan *resizing* terhadap citra *watermark* sehingga ukurannya sama seperti area yang dipilih pada langkah (b).
- (e) Proses penyisipan citra *watermark* yang telah di-*resizing* ke dalam citra *host* dengan menggunakan metode *compound mapping* sehingga menghasilkan citra ter-*watermark*.
- (f) Citra ter-*watermark* dan citra *watermark* hasil dari proses *resizing* pada langkah (d) digunakan sebagai masukan untuk proses pemulihan.
- (g) Proses pemulihan dilakukan terhadap citra ter-*watermark* juga dengan menggunakan metode *compound mapping* dan menghasilkan citra hasil yang sama dengan citra *host*.
- (h) Uji MSE dilakukan untuk membandingkan citra hasil dari proses pemulihan dengan citra *host*.

Prosedur dari sistem seperti yang telah dijelaskan di atas digambarkan dalam blok diagram pada Gambar 3.1.

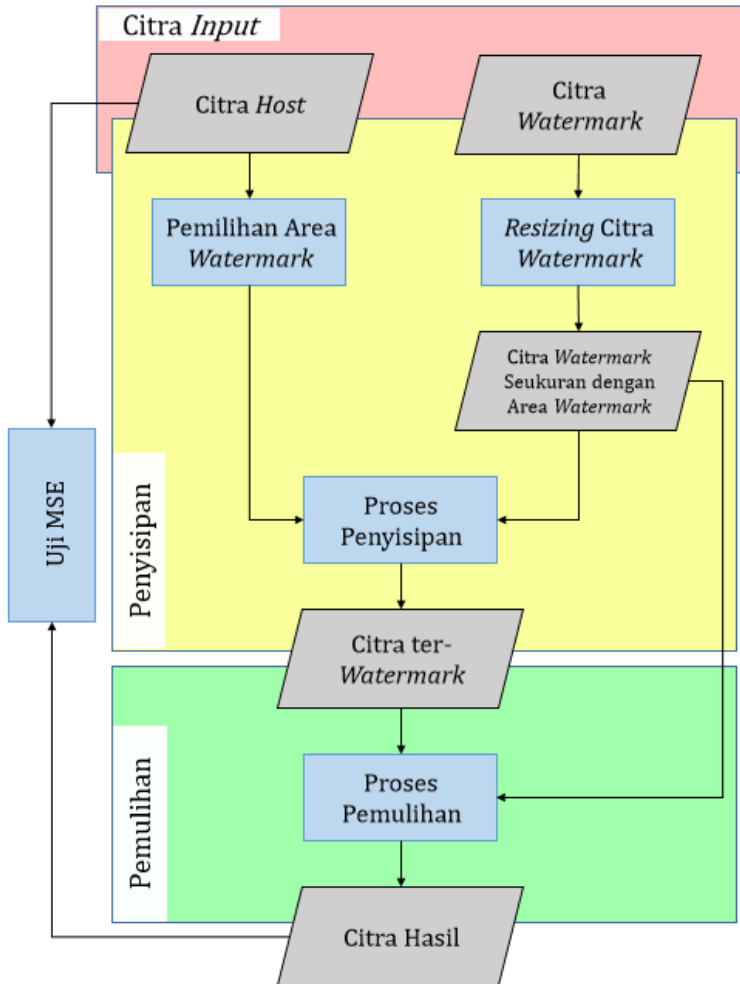
### 3. Pengujian dan Evaluasi Sistem

Pada tahap ini dilakukan penambahan *noise* terhadap citra ter-*watermark* untuk menguji kehandalan metode *watermarking* yang digunakan. Model *noise* yang akan diberikan adalah *noise salt and pepper*, yaitu terjadinya

perubahan nilai piksel menjadi maksimum atau minimum (dalam format warna 8 bit, nilai maksimum piksel adalah 255 dan nilai minimum piksel adalah 0). Model ini digunakan karena merupakan model *noise* yang umum terjadi saat transmisi data[7]. Kemudian dilakukan pengujian terhadap citra hasil pemulihan, yaitu membandingkannya dengan citra *host*(citra asli), dengan menggunakan uji MSE.

#### 4. Penarikan Kesimpulan

Tahap penarikan kesimpulan merupakan tahap akhir dalam proses penelitian Tugas Akhir ini, dimana pada tahap ini dilakukan penarikan kesimpulan terhadap hasil yang telah dicapai.



**Gambar 3.1.** Blok Diagram Watermarking Citra dengan Metode *One to one Compound Mapping*

## **BAB IV**

### **LOSSLESS VISIBLE WATERMARKING DENGAN METODE ONE TO ONE COMPOUND MAPPING**

Pada bab ini akan dibahas mengenai analisis sistem dan perancangan serta implementasi sistem meliputi pembahasan proses penyisipan citra *watermark* ke dalam citra *host* serta pemulihan citra ter-*watermark* menggunakan metode *One to One Compound Mapping*.

#### **4.1 Analisis Sistem**

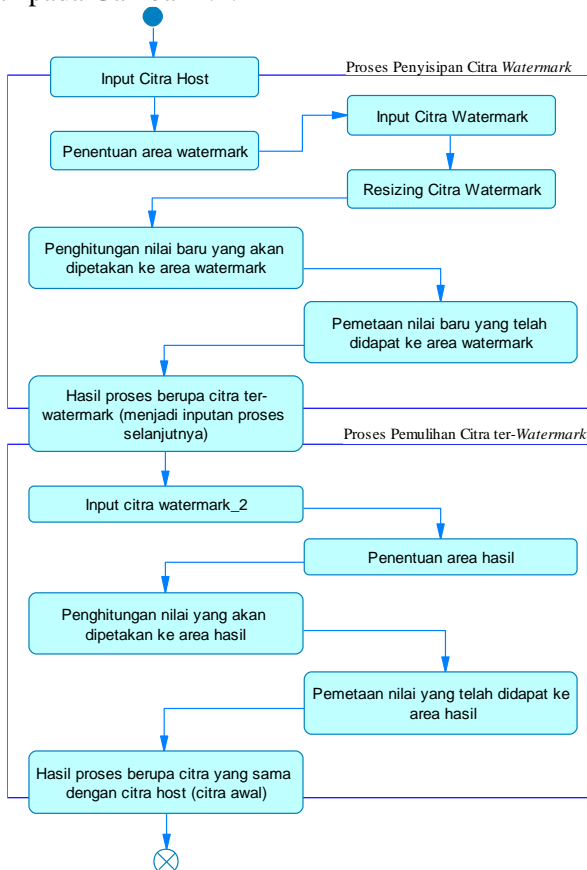
Dalam subbab ini akan dijelaskan mengenai deskripsi dari sistem yang dibangun yaitu *lossless reversible visible watermarking*, meliputi proses penyisipan citra *watermark* ke dalam citra *host* dan proses pemulihan citra ter-*watermark*.

##### **4.1.1 Deskripsi Sistem**

Sistem ini memerlukan 2 buah input citra, yaitu citra *host* dan citra *watermark*. Selanjutnya dilakukan penentuan suatu area piksel pada citra *host* yang nantinya akan disisipi dengan *watermark*. Setelah itu dilakukan *resizing* terhadap citra *watermark* agar ukurannya menjadi sama dengan ukuran area yang telah dipilih sebelumnya. Kemudian dilakukan penghitungan suatu nilai untuk dipetakan ke dalam area piksel yang telah ditentukan. Penghitungan dilakukan dengan menggunakan fungsi dari metode *One to One Compound Mapping* dengan melibatkan beberapa variabel seperti nilai piksel citra *watermark* serta nilai piksel persekitaran dari koordinat piksel citra *host* yang bersesuaian. Kemudian dilakukan pemetaan sehingga menghasilkan citra ter-*watermark*.

Proses selanjutnya yaitu pemulihan citra ter-*watermark*. Proses ini menggunakan citra *watermark* (yang telah di-*resizing*) dan citra ter-*watermark* (yang merupakan citra hasil dari proses sebelumnya) sebagai inputan. Prosesnya hampir sama seperti pada penyisipan, yaitu diawali dengan penentuan suatu area, yaitu

area dengan *watermark*, pada citra ter-*watermark*. Kemudian dilakukan penghitungan nilai piksel yang sama seperti pada proses penyisipan. Perbedaannya adalah proses penyisipan melibatkan persekitaran dari koordinat piksel citra *host*, sedangkan pada proses pemulihan yang dilibatkan adalah persekitaran koordinat piksel citra hasil pemulihan. Selanjutnya dilakukan pemetaan sehingga menghasilkan citra yang sama dengan citra *host* (citra awal). Secara umum, alir sistem ini disajikan pada Gambar 4.1.



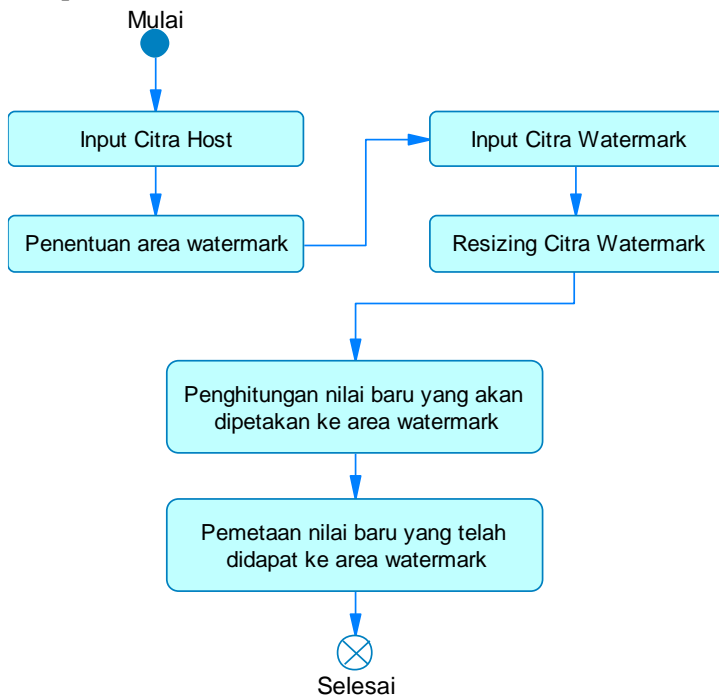
**Gambar 4.1.** Diagram Alir Sistem

## 4.2 Perancangan Sistem

Pada subbab ini dijelaskan mengenai perancangan sistem yang telah dianalisis sebelumnya. Perancangan sistem tersebut meliputi perancangan proses algoritma sistem dan perancangan antarmuka sistem.

### 4.2.1 Perancangan Penyisipan Citra *Watermark*

Perancangan proses algoritma ini secara garis besar dibagi dalam dua bagian, yaitu proses penyisipan citra *watermark* dan proses pemulihan citra ter-*watermark*.



**Gambar 4.2.** Diagram Alir Penyisipan Citra *Watermark*.

Proses pertama dari sistem ini yaitu penyisipan citra *watermark*. Diagram alir dari proses ini disajikan pada Gambar 4.2.

Penyisipan citra *watermark* memiliki beberapa tahapan sebagai berikut :

a. Pemilihan Citra *Host*

Data masukan pertama dan yang utama yaitu citra *host* yang nantinya akan disisipi dengan *watermark*. Citra *host* yang dipilih merupakan citra berwarna dengan *channel* RGB.



**Gambar 4.3.** Contoh Citra *Host*.

b. Pemilihan Area yang akan Disisipi *Watermark*

Area pada citra *host* yang akan disisipi citra *watermark* ditentukan sebelum metode utama dilakukan. Area tersebut selanjutnya dinamakan area *watermark*. Area *watermark* seluruhnya harus berada di dalam citra *host*. Ukuran maksimal area *watermark* adalah satu piksel di bawah ukuran citra *host*. Misal citra *host* berukuran  $M \times N$ , maka ukuran maksimal area *watermark* adalah  $(M - 1) \times (N - 1)$ . Sedangkan ukuran minimal dari area *watermark* tidak ditentukan karena hal tersebut relatif, tergantung pada kebutuhan area yang ingin ditutupi oleh citra *watermark*.

c. Pemilihan dan *Resizing* Citra *Watermark*

Data masukan yang kedua yaitu citra *watermark*. Sebelum melakukan proses penyisipan, citra *watermark* yang telah dipilih terlebih dahulu di-*resizing* agar ukurannya sesuai dengan ukuran area *watermark* yang ditentukan sebelumnya. Ilustrasi area *watermark* serta *resizing* citra *watermark* disajikan pada Gambar 4.4.



d. Proses Penyisipan Citra *Watermark* dengan metode *One to one Compound Mapping*

Penyisipan citra *watermark* ke dalam citra *host* dilakukan dengan metode *One to One Compound Mapping* yaitu dengan memetakan seluruh nilai piksel dari citra *watermark* ke dalam koordinat piksel yang bersesuaian pada citra *host*. Nilai piksel yang dipetakan ke citra *host* telah melalui perhitungan dengan menggunakan fungsi  $q = F_b^{-1}(F_a(p))$ , dimana  $F_a(p) = p - a$ , sehingga diperoleh

$$q = (p - a) + b \quad (4.1)$$

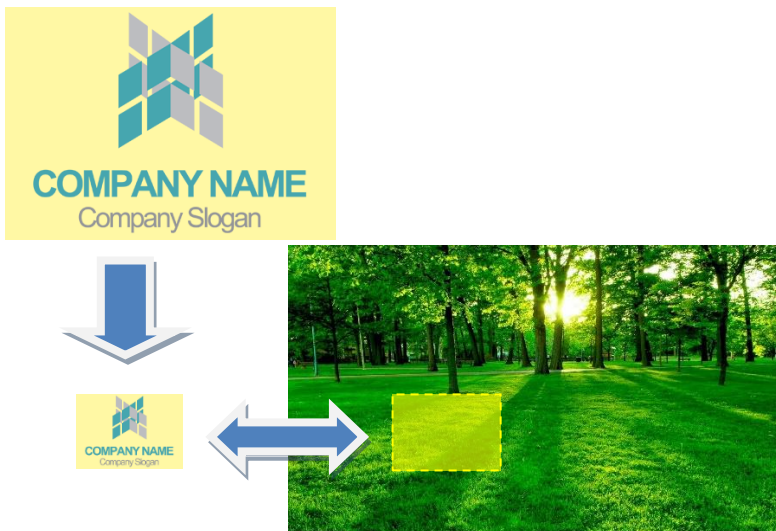
dimana :

$q$  adalah nilai piksel area *watermark* pada citra ter-*watermark*,

$p$  adalah nilai piksel area *watermark* pada citra *host*,

$a$  adalah nilai piksel persekitaran piksel  $p$ , dan

$b$  adalah nilai piksel citra *watermark*.



**Gambar 4.4.** Ilustrasi *Resizing* Citra *Watermark*.

Berikut adalah algoritma tahapan penyisipan citra *watermark*.

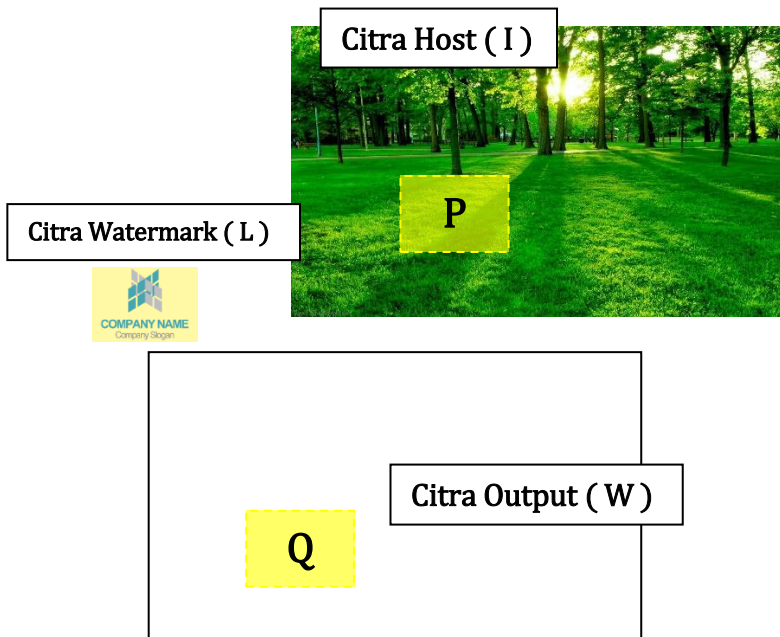
1. Siapkan citra *host*, misalkan  $I$ .
2. Pilih area *watermark* pada citra *host*, misalkan  $P$ .
3. Siapkan citra *watermark*.
4. Lakukan *resizing* citra *watermark* sehingga ukurannya menjadi satu piksel lebih dari area *watermark* yang telah ditentukan pada langkah nomer 2. Citra *watermark* hasil *resizing* dimisalkan  $L$ .
5. Simpan koordinat area *watermark* pada kelebihan piksel citra *watermark*  $L$ .
6. Buat matriks *output*, sebagai representasi dari citra ter-*watermark*, dengan ukuran yang sama dengan  $I$ , misalkan  $W$ .
7. Pilih area *watermark* pada  $W$  dengan ukuran dan posisi yang bersesuaian dengan  $P$  pada  $I$ . Area tersebut dimisalkan  $Q$ .
8. Lakukan pemetaan dari  $P$  ke  $Q$  untuk setiap pikselnya dengan fungsi  $q = F_b^{-1}(F_a(p))$ .
9. Isi area pada  $W$  yang berada di luar  $Q$  dengan nilai yang sama dengan nilai piksel yang bersesuaian pada  $I$ .

Ilustrasi citra untuk algoritma di atas dapat dilihat pada gambar 4.5.

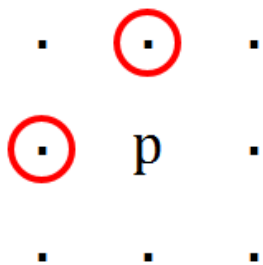
Kemudian untuk pemetaan dari  $P$  ke  $Q$ , yaitu langkah 7 algoritma sebelumnya, untuk setiap piksel citra pada  $P$ , lakukan penghitungan dengan metode *compound mapping* sesuai dengan fungsi pada Persamaan 4.1 sehingga diperoleh nilai piksel yang nantinya akan dijadikan sebagai nilai pada piksel yang bersesuaian pada  $Q$ . Penghitungan dilakukan dari piksel paling atas kiri pada area *watermark*.

Langkah penghitungan nilai yang akan dipetakan dari  $P$  ke  $Q$  adalah sebagai berikut:

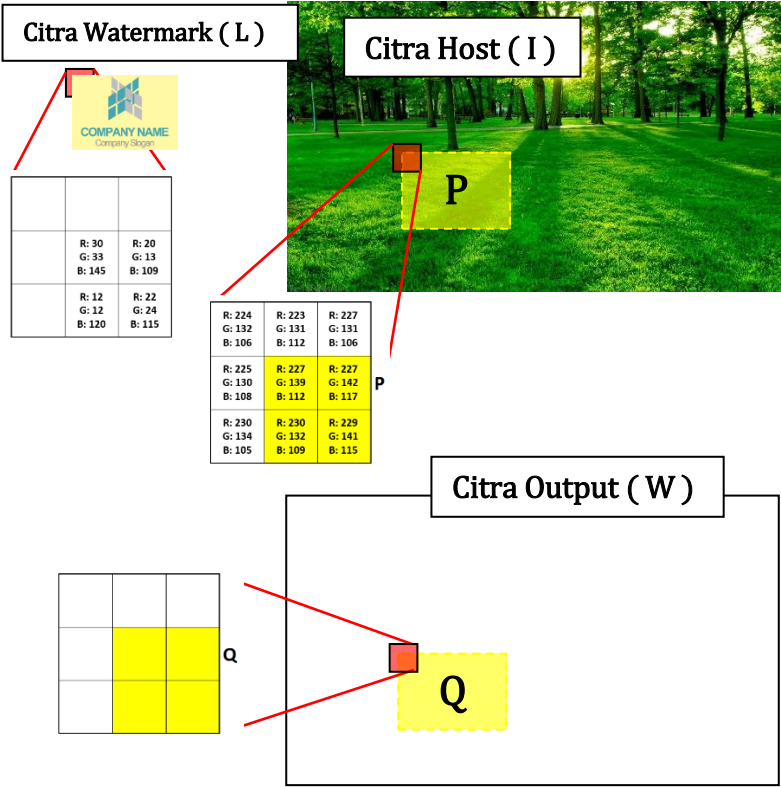
1. Tentukan nilai  $p$ , dimana  $p \in P$ .
2. Hitung nilai  $a$  berdasarkan persekitaran dari piksel  $p$ . (Ilustrasi pada Gambar 4.6)
3. Tentukan nilai  $b$  sama dengan  $l$ , dimana  $l \in L$ . ( $l$  merupakan piksel pada  $L$  yang bersesuaian dengan  $p$ )
4. Hitung nilai  $q$  dengan rumus  $q = (p - a) + b$ .
5. Ulangi langkah 1-4 untuk *channel* lainnya.



**Gambar 4.5.** Ilustrasi Citra.  $I$  adalah citra *host*.  $L$  adalah citra *watermark*.  $W$  adalah citra *ter-watermark*.

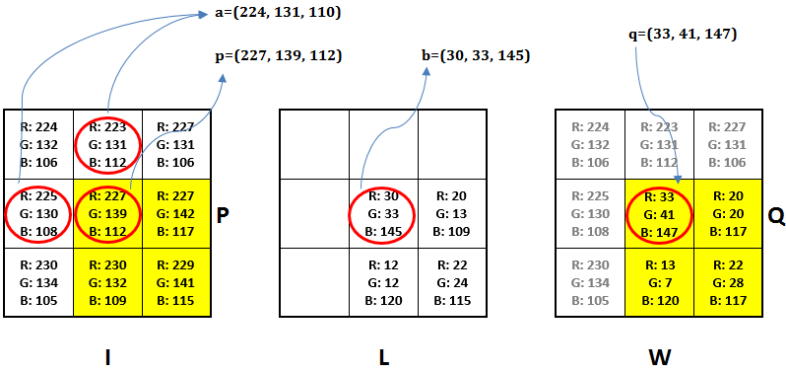


**Gambar 4.6.** Ilustrasi persekitaran piksel. Titik yang dilingkari merah merupakan persekitaran dari piksel p.



**Gambar 4.7.** Pengambilan segmen citra untuk penghitungan.





Gambar 4.9. Ilustrasi Perhitungan Nilai Piksel.

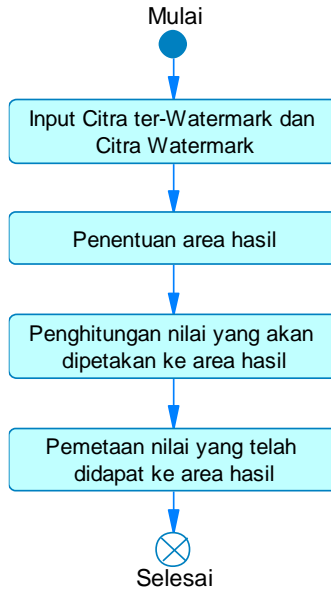
Setelah didapat nilai piksel dari area *watermark*, maka isi nilai piksel lain pada *W*, yang berada di luar area *Q*, dengan nilai yang sama dengan nilai piksel *I* yang bersesuaian. Setelah seluruh nilai piksel dari *W* diketahui, maka didapatkan citra hasil seperti pada Gambar 4.10.



Gambar 4.10. Citra ter-*Watermark* Hasil Proses Penyisipan.

4.2.2 Perancangan Pemulihan Citra Ter-*Watermark*

Proses selanjutnya dari sistem ini yaitu pemulihan citra ter-*watermark*. Secara umum Diagram dari proses ini disajikan pada Gambar 4.11.



**Gambar 4.11.** Diagram Alir Pemulihan Citra ter-*Watermark*.

Program pemulihan citra ter-*watermark* memiliki beberapa tahapan sebagai berikut :

a. Pemilihan Citra ter-*watermark* dan Citra *Watermark*

Data masukan berupa dua buah citra masing-masing sebagai citra ter-*watermark* (hasil dari proses penyisipan) dan citra *watermark* yang telah mengalami *resizing* pada proses sebelumnya.

b. Pemilihan Area Hasil

Area hasil adalah area pada citra ter-*watermark* yang mengandung *watermark* yang nantinya akan dihilangkan untuk mendapatkan hasil citra yang sama dengan citra asli (citra *host*). Dengan demikian maka ukuran area hasil akan sama dengan ukuran *watermark*.

c. Proses Pemulihan Citra ter-*Watermark*

Proses ini merupakan kebalikan dari proses penyisipan. Seluruh nilai piksel dari citra *watermark* akan dipetakan ke area

hasil pada citra ter-*watermark*. Fungsi pemetaan yang digunakan pada proses ini merupakan invers dari fungsi pemetaan pada proses penyisipan yaitu  $p = F_a^{-1}(F_b(q))$ . Sehingga dengan melakukan invers pada Persamaan 4.1, diperoleh fungsi pemetaan

$$p = (q - b) + a \quad (4.2)$$

dimana :

$p$  adalah nilai piksel area bekas *watermark* pada citra hasil pemulihan,

$q$  adalah nilai piksel area *watermark* pada citra ter-*watermark*,

$a$  adalah nilai piksel persekitaran piksel  $p$ , dan

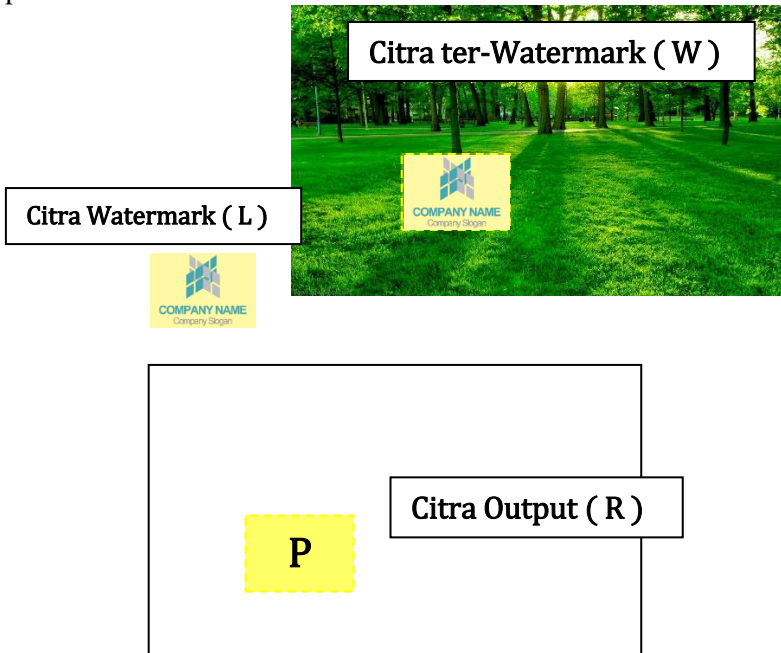
$b$  adalah nilai piksel citra *watermark*.

Berikut adalah algoritma tahapan pemulihan citra ter-*watermark*.

1. Siapkan citra ter-*watermark*, misalkan  $W$ .
2. Siapkan citra *watermark* yang telah mengalami *resizing* pada proses sebelumnya, misalkan  $L$ .
3. Pilih area hasil pada citra ter-*watermark* berdasarkan koordinat yang disimpan pada  $L$ , misalkan  $Q$ .
4. Buat matriks *output*, sebagai representasi dari citra hasil pemulihan, dengan ukuran yang sama dengan  $W$ , misalkan  $R$ .
5. Pilih area pada  $R$  dengan ukuran dan posisi yang bersesuaian dengan  $Q$  pada  $W$ . Area tersebut akan menjadi area bekas *watermark*, dimisalkan  $P$ .
6. Isi koordinat piksel pada  $R$  yang berada di luar area  $P$  dengan nilai yang sama dengan nilai piksel pada koordinat piksel  $W$  yang bersesuaian. (Hasilnya seperti pada Gambar 4.13)
7. Lakukan pemetaan dari  $Q$  ke  $P$  untuk setiap pikselnya dengan fungsi  $p = F_a^{-1}(F_b(q))$ .



Ilustrasinya citra untuk algoritma pemulihan dapat dilihat pada Gambar 4.12.

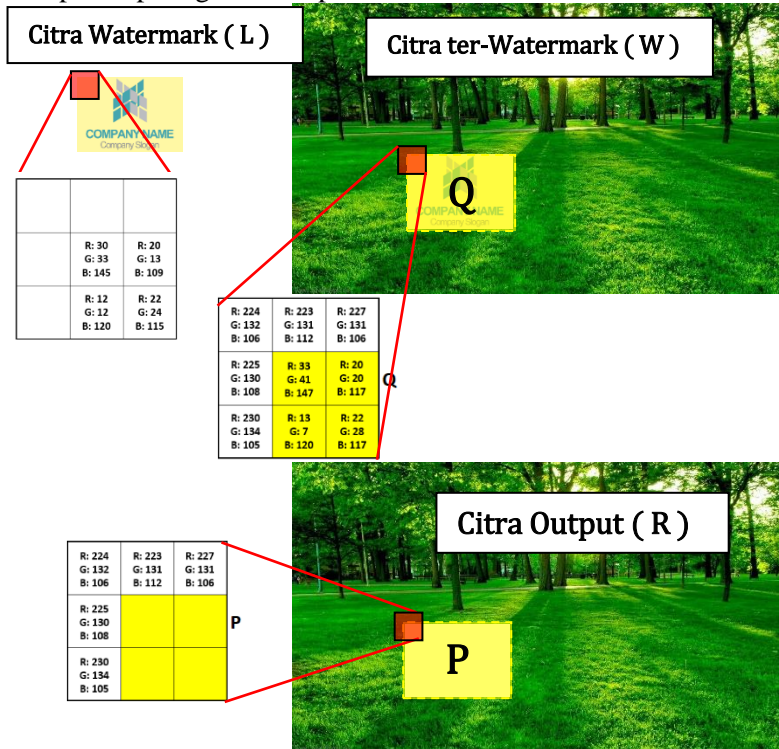


**Gambar 4.12.** Ilustrasi Citra.  $W$  adalah citra ter-watermark.  $L$  adalah citra watermark.  $R$  adalah citra output (hasil).



**Gambar 4.13.** Hasil Pemetaan Nilai Piksel di Luar Area *Watermark*.

Selanjutnya untuk pemetaan dari  $Q$  ke  $P$ , langkah ke-7 dari algoritma sebelumnya, untuk setiap piksel citra pada  $Q$ , lakukan penghitungan dengan metode *compound mapping* sehingga diperoleh nilai citra yang nantinya akan dijadikan sebagai nilai pada piksel yang bersesuaian pada  $P$ . Penghitungan dilakukan dari piksel paling atas kiri pada area *watermark*.



**Gambar 4.14.** Pengambilan segmen citra untuk penghitungan.

Langkah penghitungan nilai piksel  $P$  sebagai berikut:

1. Tentukan nilai  $q$ , dimana  $q \in Q$ .
2. Hitung nilai  $a$  berdasarkan persekitaran dari piksel  $p$  yang akan dicari nilainya. (Ilustrasi persekitaran pada

Figure 1 illustrates the structure of the input data for the model. It consists of three 3x3 grids labeled W, L, and R, representing different views or features. The grids are arranged horizontally, with W on the left, L in the middle, and R on the right. Arrows indicate a flow from W to L and from L to R.

**Grid W (Left):**

R: 224 G: 132 B: 106	R: 223 G: 131 B: 112	R: 227 G: 131 B: 106
R: 225 G: 130 B: 108	<b>R: 33 G: 41 B: 147</b>	R: 20 G: 20 B: 117
R: 230 G: 134 B: 105	R: 13 G: 7 B: 120	R: 22 G: 28 B: 117

**Grid L (Middle):**

	R: 30 G: 33 B: 145	R: 20 G: 13 B: 109
	R: 12 G: 12 B: 120	R: 22 G: 24 B: 115

**Grid R (Right):**

R: 224 G: 132 B: 106	R: 223 G: 131 B: 112	R: 227 G: 131 B: 106
R: 225 G: 130 B: 108		
R: 230 G: 134 B: 105		

Misal perhitungan pertama dilakukan pada *channel* piksel *red*, maka  $q = 33$ . Kemudian hitung nilai  $a$  berdasarkan persekitaran (nilai piksel atas dan kiri) dari  $p$

Kemudian isi nilai  $b$  sama dengan  $l$ , dimana  $l \in L$ , untuk piksel yang bersesuaian dengan  $p$ , maka

Kemudian lakukan pemetaan  $p = F_a^{-1}(F_b(q))$ , dimana fungsi  $F_a(p) = p - a$ , sehingga

$p = 227$

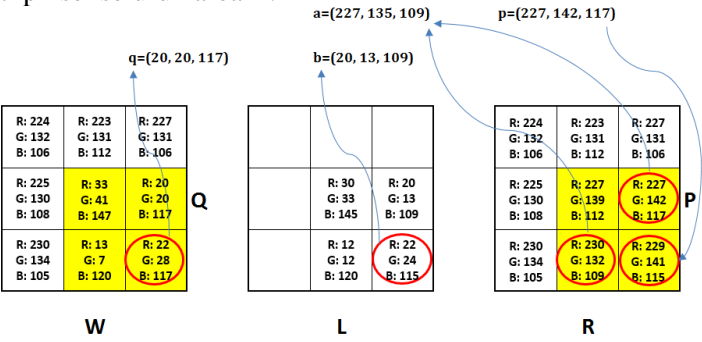
Ulangi langkah yang sama untuk menghitung *channel green* dan *blue*, sehingga diperoleh nilai  $p = (227, 139, 112)$ .

R: 224 G: 132 B: 106	R: 223 G: 131 B: 112	R: 227 G: 131 B: 106
R: 225 G: 130 B: 108	R: 227 G: 139 B: 112	
R: 230 G: 134 B: 105		

P

**Gambar 4.16.** Citra *R* setelah ditentukan nilai piksel pertama pada area hasil *P*.

Untuk menentukan nilai piksel pada koordinat selanjutnya, ulangi langkah yang dilakukan sebelumnya dengan menggunakan nilai piksel yang baru didapat sebagai persekitaran untuk mencari nilai  $a$ . Ulangi langkah tersebut untuk menghitung nilai piksel seluruh area *P*.



**Gambar 4.17.** Ilustrasi perhitungan nilai piksel



**Gambar 4.18.** Contoh Citra Hasil Pemulihan.



Karena *range* nilai piksel antara 0 sampai 255, jika nilai  $q$  tersebut dipetakan ke dalam matriks citra, maka nilai piksel pada koordinat tersebut akan menjadi 255. Hasil pemetaan ke area *watermark* bisa dilihat pada Gambar 4.20.

R: 240	R: 253	R: 246				R: 240	R: 253	R: 246
R: 253	R: 255	R: 250	P		R: 255	R: 253		Q
R: 238	R: 251	R: 249			R: 253	R: 250		
I				L				W

**Gambar 4.20** Nilai Piksel Setelah Dilakukan Pemetaan dari  $P$  ke  $Q$

Kemudian terhadap citra ter-*watermark* tersebut dilakukan proses pemulihan. Lakukan langkah-langkahnya seperti pada proses pemulihan yang telah dijelaskan sebelumnya sampai pada pemetaan diluar area hasil, hingga didapatkan nilai piksel seperti pada Gambar 4.21.

R: 240	R: 253	R: 246				R: 240	R: 253	R: 246
R: 253	R: 255	R: 252	Q		R: 255	R: 253		P
R: 238	R: 255	R: 248			R: 253	R: 250		
W				L				R

**Gambar 4.21** Nilai Piksel Setelah Dilakukan Pemetaan di Luar Area Hasil

Selanjutnya lakukan perhitungan untuk memetakan  $Q$  ke  $P$ . Ditentukan  $q = 255$ . Lalu cari nilai  $a$  berdasarkan persekitaran  $p$  yang akan dicari, didapat  
 $a = 253$ .

Kemudian isi nilai  $b$  sama dengan  $l$ , dimana  $l \in L$ , untuk piksel yang bersesuaian dengan  $p$ , maka

$$b = 255$$

Kemudian lakukan pemetaan  $p = F_a^{-1}(F_b(q))$ , dimana fungsi  $F_a(p) = p - a$ , sehingga

$$p = F_a^{-1}(F_b(q))$$

$$p = (q - b) + a$$

$$p = (255 - 255) + 253$$

$$p = 253$$

Setelah pemetaan selesai, hasil nilai pikselnya akan seperti pada Gambar 4.22.

R: 240	R: 253	R: 246		R: 240	R: 253	R: 246
R: 253	R: 255	R: 252	Q		R: 255	R: 253
R: 238	R: 255	R: 248			R: 253	R: 250
W				L		
				R		

**Gambar 4.22** Nilai Piksel Setelah Dilakukan Pemetaan dari  $Q$  ke  $P$

R: 240	R: 253	R: 246		R: 240	R: 253	R: 246
R: 253	R: 255	R: 250	P	R: 253	R: 253	R: 249
R: 238	R: 251	R: 249		R: 238	R: 255	R: 250
I				R		

**Gambar 4.23** Nilai Piksel Citra *Host* ( $I$ ) dan Citra Hasil Pemulihan ( $R$ )

Seperti yang ditampilkan pada Gambar 4.23, kasus seperti ini dapat mengakibatkan citra hasil pemulihan yang didapat tidak sama seperti citra *host* sesuai dengan yang diharapkan. Untuk mengatasi hal ini, diperlukan suatu algoritma tambahan. Berikut ini adalah algoritma proses penyisipan yang telah ditambahkan dengan algoritma yang diperlukan.

1. Tentukan nilai  $p$ , dimana  $p \in P$ .
2. Hitung nilai  $a$  berdasarkan persekitaran dari piksel  $p$ .  
(Ilustrasi pada Gambar 4.6)
3. Tentukan nilai  $b$  sama dengan  $l$ , dimana  $l \in L$ . ( $l$  merupakan piksel pada  $L$  yang bersesuaian dengan  $p$ )
4. Hitung nilai  $q$  dengan rumus  $q = (p - a) + b$ .  
Jika  $q > 255$  maka hitung  $l - (q - 255)$ , kemudian hasilnya simpan ke dalam  $l$  dan isi  $q = 255$ .  
Jika  $q < 0$  maka hitung  $l + (0 - q)$ , kemudian hasilnya simpan ke dalam  $l$  dan isi  $q = 0$ .  
Jika  $0 \leq q \leq 255$  maka nilai  $l$  tetap.
5. Ulangi langkah 1-4 untuk *channel* lainnya.

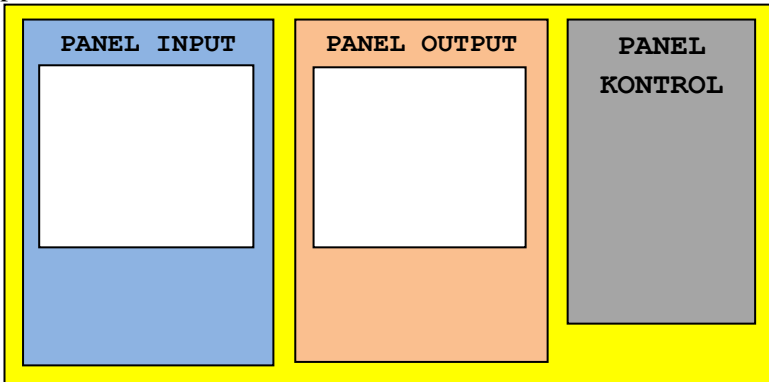
Selanjutnya untuk proses pemulihan, lakukan dengan algoritma yang sama dengan algoritma pemulihan sebelumnya.

#### 4.2.4 Perancangan Antar Muka Sistem

Untuk mempermudah eksperimen, dibutuhkan antarmuka dari sistem yang telah diancang. Terdapat 2 buah halaman yaitu halaman proses penyisipan dan halaman proses pemulihan. Pada halaman penyisipan dilakukan pemilihan citra *host* dan citra *watermark*. Kemudian dilakukan proses penyisipan citra *watermark* ke dalam citra *host*. Kemudian pada halaman pemulihan dilakukan proses pemilihan citra ter-*watermark* yang akan dipulihkan, serta citra *watermark*. Kemudian dilakukan proses pemulihan untuk mendapatkan citra hasil pemulihan yang diharapkan akan sama dengan citra *host* yang ada pada proses



sebelumnya. Rancangan kedua halaman tersebut ditunjukkan pada Gambar 4.24



**Gambar 4.24** Antarmuka Halaman Program *Watermarking*.

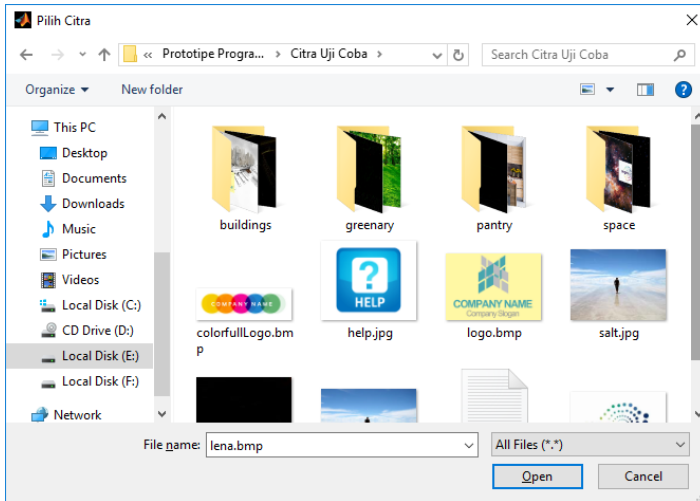
## 4.3 Implementasi Sistem

### 4.3.1 Implementasi *Input* Citra

*Input* citra terdapat dalam kedua proses baik penyisipan maupun pemulihan. Pada tahap ini *user* diharuskan memilih citra yang akan digunakan dalam masing-masing proses, kemudian sistem akan menampilkan citra yang dipilih tersebut. Proses tersebut diimplementasikan dalam program berikut :

```
[filename,path]=uigetfile('*.','Pilih
Citra');
i=imread(strcat(path,filename));
axes(handles.axes2);
imshow(i);
ir=i(:,:,1);
ig=i(:,:,2);
ib=i(:,:,3);
```

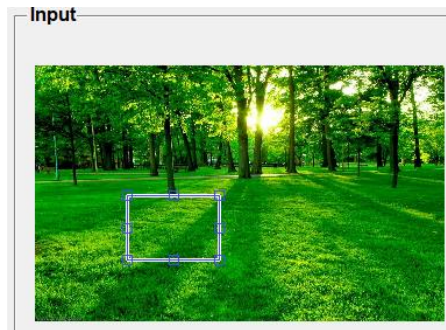
Gambar 4.25 adalah tampilan antar muka pengambilan *input* citra.



**Gambar 4.25** Tampilan saat pemilihan citra *input*.

#### 4.3.2 Implementasi Pemilihan Area *Watermark*

Sebelum proses penyisipan, perlu dilakukan pemilihan area yang akan disisipi *watermark*. Setelah *user* melakukan *input*, sistem akan menampilkan citra *input* tersebut, kemudian *user* melakukan *click* dan *drag* pada panel citra *host* yang secara langsung akan diikuti oleh kotak biru. Area dalam kotak biru tersebut merupakan area *watermark*.



**Gambar 4.26** Tampilan saat pemilihan area *watermark*.

### 4.3.3 Implementasi Proses *Resizing* Citra *Watermark*

Setelah *user* memilih area *watermark* pada citra *host*, sistem akan mengambil nilai panjang dan lebar kotak biru pada panel citra *host*. Kemudian sistem menyamakan ukuran citra *watermark* dengan ukuran kotak biru tersebut. Citra *watermark* yang berukuran sama dengan kotak biru tersebut yang akan menjadi bahan untuk proses penyisipan.

Berikut kode program untuk *resizing* citra *watermark*.

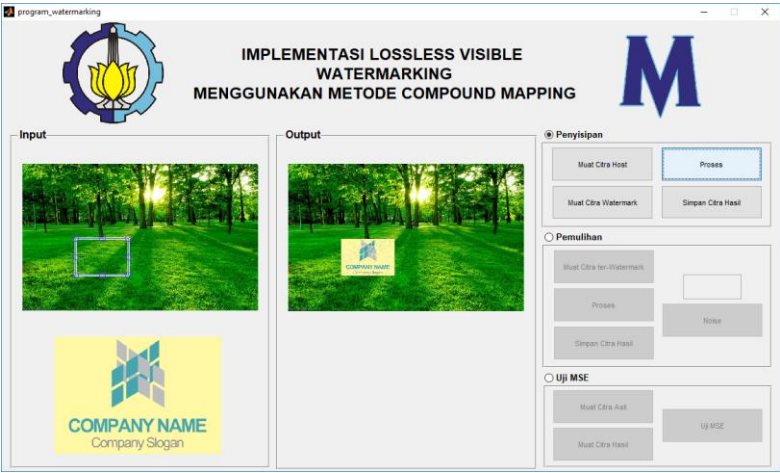
```
pos=getPosition(h);
x=ceil(pos(2));
y=ceil(pos(1));
width=ceil(pos(3));
height=ceil(pos(4));

l = imresize(l,[height width]);
lr=l(:,:,1);
lg=l(:,:,2);
lb=l(:,:,3);
```

### 4.3.4 Implementasi Proses Penyisipan Citra *Watermark*

```
ar1 =ir(e,f-1);
ar2 =ir(e-1,f);
ar =uint8(mean([ar1;ar2]));
pr =ir(e,f);
br =l2r(e,f);
qr =(int16(pr)-int16(ar))+int16(br);
if qr>255
    keyr(e-(y-1),f-(x-1))=int16(lr(e-(y-1),f-(x-1)))-(qr-255);
elseif qr<0
    keyr(e-(y-1),f-(x-1))=int16(lr(e-(y-1),f-(x-1)))+(0-qr);
else
    keyr(e-(y-1),f-(x-1))=int16(lr(e-(y-1),f-(x-1)));
end
wr(e,f) = qr;
```

Gambar 4.27 adalah tampilan antar muka proses penyisipan *watermark*.



**Gambar 4.27** Implementasi proses penyisipan citra *watermark*.

### 4.3.5 Implementasi Pemilihan Area Hasil

Pada tahap ini sistem akan secara otomatis memunculkan sebuah kotak sebagai area hasil setelah sistem menampilkan citra *input*-an. Pada tahap ini user tidak perlu melakukan apapun karena kotak area hasil tersebut sudah berada pada posisi yang tepat.

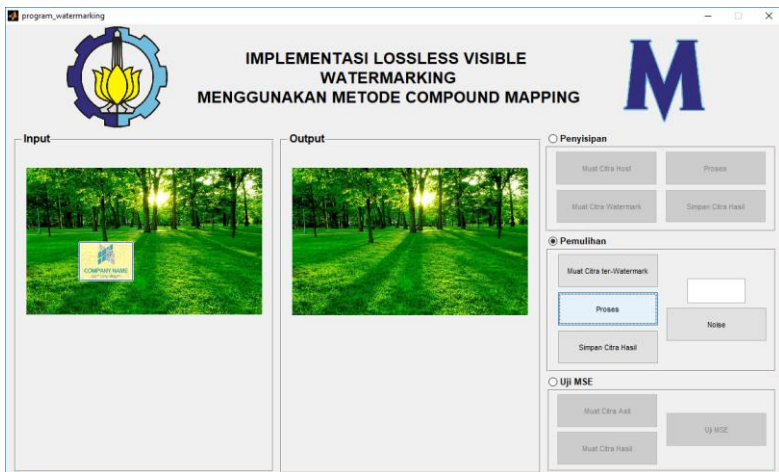


**Gambar 4.28** Tampilan saat pemilihan area hasil.

#### 4.3.6 Implementasi Proses Pemulihan Citra ter-Watermark

Berikut kode dari proses ini:

```
ar1 =rr(e,f-1);
ar2 =rr(e-1,f);
ar =uint8(mean([ar1;ar2]));
qr =wr(e,f);
br =l2r(e,f);
pr =(int16(qr)-int16(br))+int16(ar);
rr(e,f) =pr;
```



**Gambar 4.29** Antar muka proses pemulihan citra ter-watermark.



## **BAB V**

### **UJI COBA DAN ANALISIS HASIL**

Pada bab ini dijelaskan mengenai uji coba terhadap aplikasi yang telah dibuat serta pembahasan tentang hasil dari uji coba aplikasi tersebut. Uji coba dilakukan terhadap beberapa citra dengan tingkat kompleksitas warna yang berbeda-beda. Terdapat dua uji coba yang akan dilakukan yaitu, uji coba penambahan *watermark* dan pemulihan tanpa menambahkan *noise* pada citra yang ter-*watermark* dan uji coba penambahan *watermark* dan pemulihan dengan menambahkan *noise* pada citra yang ter-*watermark*. Hal ini dimaksudkan untuk menguji kehandalan proses pemulihan terhadap citra ter-*watermark* yang telah diberi *noise* untuk kemudian akan diuji performansinya. Pengujian citra *host* dengan citra hasil pemulihan dilakukan menggunakan uji MSE.






#### **5.1 Uji Coba Terhadap Citra Tanpa Noise**

Uji coba pada subbab ini dilakukan tanpa menambahkan *noise* pada citra yang diuji. Pengujian akan dilakukan terhadap beberapa citra yang memiliki tingkat kompleksitas warna yang berbeda-beda. Masing-masing citra akan diuji menggunakan tiga citra *watermark* yang kompleksitas warnanya juga berbeda-beda, sehingga setiap citra akan diuji coba sebanyak tiga kali. Data-data citra *host* dan citra *watermark* masing-masing diberikan pada Tabel 5.1 dan Tabel 5.2.

Skenario pengujiannya adalah sebagai berikut:




- Menentukan citra *host* yang akan di *watermark*
- Menentukan citra *watermark*
- Melakukan penyisipan citra *watermark* ke dalam citra *host* sehingga menghasilkan citra ter-*watermark*
- Melakukan pemulihan pada citra ter-*watermark*
- Menguji citra *host* dengan citra hasil pemulihan menggunakan MSE.

**Tabel 5.1** Data Citra *Host* yang Digunakan.

No	Nama	Citra <i>Host</i>	
1	greenary.bmp		
2	pantry.bmp		
3	buildings.png		
4	denahRumah.jpg		
5	desk.jpg		












**Tabel 5.2** Data Citra *Watermark* yang Digunakan.







No	Nama	Citra <i>Watermark</i>
1	logo.png	
2	colourBoxSquare.png	
3	adingAttamimi.png	

Berikut ini adalah hasil penyisipan Citra *Watermark*.

**Tabel 5.3** Data Citra ter-*Watermark*.

No	Citra <i>Host</i>	Citra <i>Watermark</i>	Citra ter- <i>Watermark</i>
1			
2			
3			

4			
5			
6			
7			
8			
9			













10			
11			

Berdasarkan hasil yang ditunjukkan pada Tabel 5.3, dapat dilihat bahwa, metode *Compound Mapping* mampu menangani proses *watermarking* dengan baik di semua tingkat kompleksitas citra yang digunakan.

Selanjutnya akan dilakukan pemulihan terhadap citra ter-*watermark* yang telah didapat. Kemudian citra hasil pemulihan akan dibandingkan dengan citra *host* melalui uji MSE. Data citra hasil pemulihan dan nilai MSE-nya ditampilkan dalam Tabel 5.4.

**Tabel 5.4** Data Citra Hasil Pemulihan dan Nilai MSE.

No	Citra ter-Watermark	Citra Hasil Pemulihan	MSE
1			0
2			0

3			0
4			0
5			0
6			0
7			0
8			0

9			0
10			33,5384 01
11			19,5780 65

Berdasarkan hasil yang ditunjukkan pada Tabel 5.4, dapat dilihat bahwa citra hasil pemulihan pada seluruh citra BMP dan PNG adalah sama dengan citra *host*. Hal tersebut didukung dengan nilai MSE yang sama dengan nol. Sedangkan hasil pemulihan citra JPG tidak sama dengan citra *host*.

## 5.2 Uji Coba Terhadap Citra dengan Noise

Pada subbab ini akan dilakukan pengujian dengan menambahkan *noise* pada citra ter-*watermark*. Skenario pengujian ini adalah:

- Menentukan citra *host* yang akan di *watermark*
- Menentukan citra *watermark*
- Melakukan penambahan citra *watermark* ke dalam citra *host* sehingga menghasilkan citra ter-*watermark*
- Menambahkan *noise* pada citra ter-*watermark*
- Melakukan pemulihan pada citra ter-*watermark*

Citra ter-*watermark* yang digunakan dalam pengujian *noise* ini adalah citra ter-*watermark* yang bernomor 7 pada Tabel 5.3.





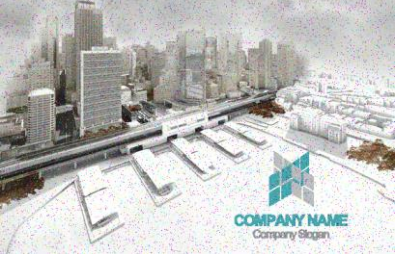
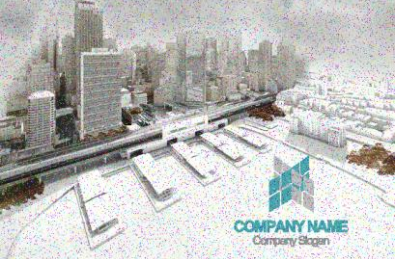
Citra tersebut ditampilkan kembali pada Gambar 5.1. Pengujian dilakukan yaitu dengan memberi noise *salt and pepper* dengan parameter densitas *noise* yang berbeda-beda. Dari percobaan, diberikan 7 macam densitas noise dengan nilai terendah sebesar 0,001 dan nilai tertinggi sebesar 0,1. Hasil pemberian *noise* pada citra ter-watermark ditunjukkan melalui Tabel 5.5.




**Gambar 5.1** Citra buildings.png ter-Watermark

**Tabel 5.5** Data Citra ter-Watermark ber-Noise.


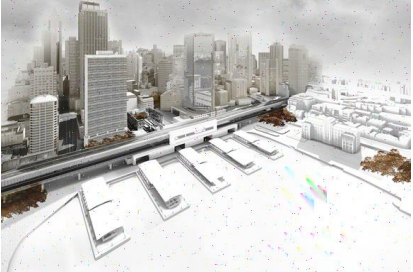
No	Densitas Noise	Citra ter-Watermark ber-Noise
1	0,001	
2	0,004	

3	0,007	 <p>COMPANY NAME Company Slogan</p>
4	0,01	 <p>COMPANY NAME Company Slogan</p>
5	0,04	 <p>COMPANY NAME Company Slogan</p>
6	0,07	 <p>COMPANY NAME Company Slogan</p>

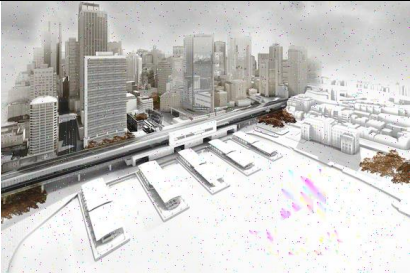
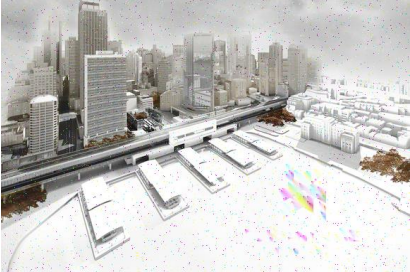
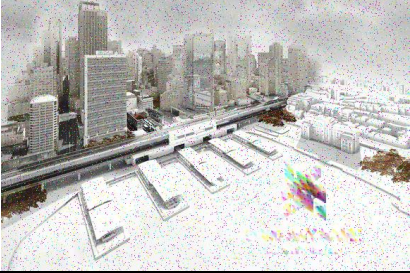
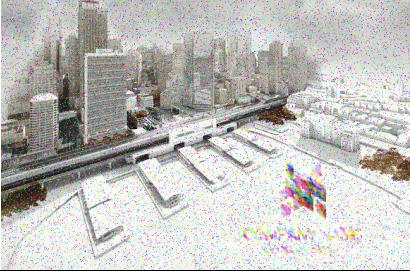
7	0,1	
---	-----	---

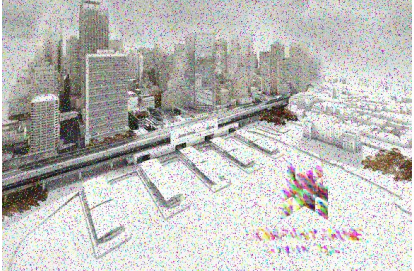
Dari citra ter-watermark yang telah ditambah *noise* yang ditunjukkan pada Tabel 5.5, dilakukan proses pemulihan citra ter-watermark untuk mengetahui performanya. Kemudian citra hasil pemulihan tersebut dibandingkan dengan citra asli melalui uji MSE. Hasil pemulihan citra ter-watermark dan nilai MSE ditunjukkan melalui Tabel 5.6

**Tabel 5.6** Data Citra Hasil Pemulihan dari Citra ber-Noise dan Nilai MSE.

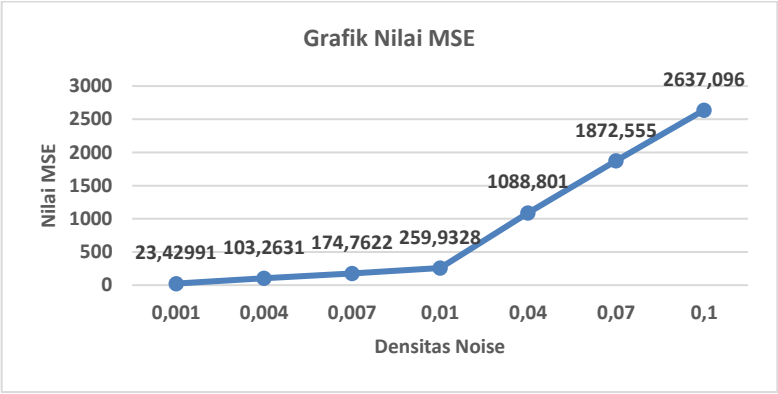
No	Densitas Noise	Citra Hasil Pemulihan	MSE
1	0,001		23,42991
2	0,004		103,2631



3	0,007		174,7622
4	0,01		259,9328
5	0,04		1088,801
6	0,07		1872,555

7	0,1		2637,096
---	-----	---	----------

Berdasarkan hasil pemulihan citra ber-*noise* pada Tabel 5.6, dapat diketahui bahwa semakin besar tingkat *noise* yang diberikan pada citra ter-*watermark*, maka semakin buruk kualitas citra hasil pemulihannya. Grafik nilai MSE terhadap perubahan densitas *noise* pada tiap citra yang diuji ditunjukkan pada Gambar 5.2.



**Gambar 5.2** Grafik Nilai MSE Pada Uji Citra ber-*Noise*

## BAB VI

### KESIMPULAN DAN SARAN

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. Di samping itu, pada bab ini juga dimasukkan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

#### 6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian yang telah dilakukan terhadap sistem maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Pengimplementasian *reversible watermarking* pada citra digital menggunakan metode *compound mapping* terbagi menjadi 2 proses utama, yaitu penyisipan dan pemulihan. Proses penyisipan citra *watermark* dilakukan menggunakan rumus  $q = (p - a) + b$ . Proses pemulihan citra ter-*watermark* dilakukan menggunakan rumus  $p = (q - b) + a$ .
2. Pada proses penyisipan citra *watermark*, metode ini mampu menangannya dengan baik untuk semua format citra yang diuji, yaitu BMP, PNG, dan JPG.
3. Pada pemulihan citra ter-*watermark* yang tidak diberi *noise*, hasil perhitungan MSE menghasilkan nilai sama dengan 0 (nol) untuk citra dengan format BMP dan PNG, sehingga dapat dikatakan bahwa citra hasil pemulihannya sama persis dengan citra *host*.
4. Pada pemulihan citra ter-*watermark* tanpa *noise* yang berformat JPG, citra hasil pemulihan tidak sama persis dengan citra *host*. Hal tersebut karena terjadinya kompresi yang *lossy* (perubahan nilai piksel) saat penyimpanan citra JPG ter-*watermark*, sehingga nilainya tidak sesuai dengan perhitungan. Karena metode yang digunakan adalah pemetaan satu-satu, apabila nilai piksel yang ingin dipulihkan nilainya tidak sesuai, maka nilai piksel hasil

pemulihannya pun tidak akan sesuai dengan yang diinginkan, yaitu sama persis dengan nilai piksel awal.

## **6.2 Saran**

Berdasarkan hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya yaitu:

1. Karena fungsi yang digunakan dalam metode ini sudah dipublikasikan, maka tingkat keamanannya menjadi kurang baik. Sehingga untuk penelitian selanjutnya alangkah lebih baik jika ditambahkan metode lain untuk meningkatkan keamanannya.
2. Pada penelitian ini, hanya dilakukan uji coba terhadap citra BMP, PNG, dan JPG. Sehingga akan lebih baik jika pada penelitian selanjutnya dilakukan uji coba terhadap citra dengan format yang lain.
3. Untuk penelitian selanjutnya, perlu dilakukan pengujian dengan metode yang lain untuk menguji kehandalan metode ini.

## DAFTAR PUSTAKA

- [1] Boyat, AK. Joshi BK. (2015). “*A Review Paper: Noise Models in Digital Image Processing*”. Jurnal. Signal & Image Processing : An International Journal (SIPIJ) Vol 6 No 2.
- [2] Chandramouli, R. Memon, Nasir. Rabbani, Majid. (2002). “*Digital Watermarking*”. Jurnal. Encyclopedia of Imaging Science and Technology.
- [3] Gonzales, RC. Woods, RE. (2002). “*Digital Image Processing*”. New Jersey : Prentice Hall, Inc.
- [4] Sundeep, K. Swamy, P.M. (2013). “*Generic Lossless Visible Watermarking*”. Jurnal. International Journal of Engineering Sciences & Management. Vol 3.
- [5] Tian, Jun. (2002). “*Reversible Watermarking by Difference Expansion*”. Paper. Digimarc Corporation. USA.
- [6] Tian, Jun. (2002). “*Wavelet-based Reversible Watermarking for Authentication*”. Paper. Digimarc Corporation. USA.
- [7] <http://www.click3d.co.uk/portfolio>, diakses pada tanggal 22 Juli 2016 pukul 17.00 WIB.



## BIODATA PENULIS



Penulis memiliki nama lengkap Ginanjar Dwi Erdityawan, lahir di Probolinggo pada tanggal 15 April 1992. Penulis berasal dari Kota Banyuwangi, bertempat tinggal di Perumahan Griya Giri Mulya Blok VW-23 Kel. Klatak, Kec. Kalipuro, Kab. Banyuwangi. Pendidikan formal yang pernah ditempuh yaitu SD Negeri 1 Lateng-Banyuwangi, SMP Negeri 1 Banyuwangi, dan SMA Negeri 1 Glagah-Banyuwangi. Kemudian, penulis melanjutkan studi di jurusan Matematika ITS, dengan bidang minat ilmu komputer. Dalam bidang minat ini penulis mengenal beberapa bahasa pemrograman diantaranya adalah C, C++, Java, PHP-MySQL, dan MATLAB. Di akhir masa-masa studi S-1, penulis juga aktif menjadi asisten dosen di PAPSI-ITS. Selama penulisan Tugas Akhir ini Penulis tidak lepas dari kekurangan, untuk itu penulis mengharapkan kritik dan saran. Apabila ada pertanyaan mengenai Tugas Akhir ini, silakan bertanya kepada penulis. Kritik, saran, dan pertanyaan dapat dikirimkan melalui *e-mail* ke [ginanjarerdityawan@gmail.com](mailto:ginanjarerdityawan@gmail.com)

